

Pertanyaan Teori

1. Apa perbedaan antara algoritma dan program?

Solution: Algoritma berfungsi untuk memberikan solusi masalah dalam bentuk Langkah-langkah. Sedangkan program berfungsi untuk menjalankan dari solusi algoritma.

2. Apakah yang dimaksud dengan algoritma?

Solution: Algoritma merupakan deskripsi yang menjelaskan apa yang harus dikerjakan. Algoritma bisa disebut juga yaitu resep, langkahnya seperti "*Panggang sampai selesai*" tidak jelas karena tidak menjelaskan apa yang berarti "*selesai*".

3. Apa yang dimaksud dengan flowchart dan pseudocode?

Solution: Flowchart adalah sebuah cara merepresentasikan algoritma dengan menggunakan simbol-simbol geometris dan grafik. Sedangkan Pseudocode adalah sebuah cara merepresentasikan algoritma dengan menggunakan bahasa sehari-hari

4. Sebutkan fungsi desk checking?

Solution: Desk checking adalah suatu teknik manual yang digunakan untuk memeriksa logika dari suatu algoritma. Manfaatnya untuk melakukan proses pengecekan algoritma sebelum membuat program untuk mengurangi kesalahan.

5. Suatu algoritma terdiri dari tiga struktur dasar, yaitu runtunan, pemilihan, dan perulangan. Jelaskan masing-masing!

Solution:

1. Algoritma runtunan merupakan sekumpulan perintah atau pernyataan yang dikerjakan komputer berdasarkan dengan urutan perintahnya.
2. Algoritma perulangan adalah proses yang digunakan untuk mengulang beberapa perintah sesuai dengan jumlah yang telah ditentukan.
3. Algoritma pemilihan adalah proses yang dilakukan sesuai dengan persyaratan atau kondisi tertentu yang sudah terpenuhi.

6. Kapan modularisasi dibutuhkan dalam pembuatan algoritma?

Solution: Ketika source code program sudah terlalu besar dan kompleks, maka dapat dipecahkan menjadi beberapa modul. Fungsi modul supaya kita membacanya lebih tertata. Dengan begitu, bisa diimplementasikan jauh lebih mudah dipahami.

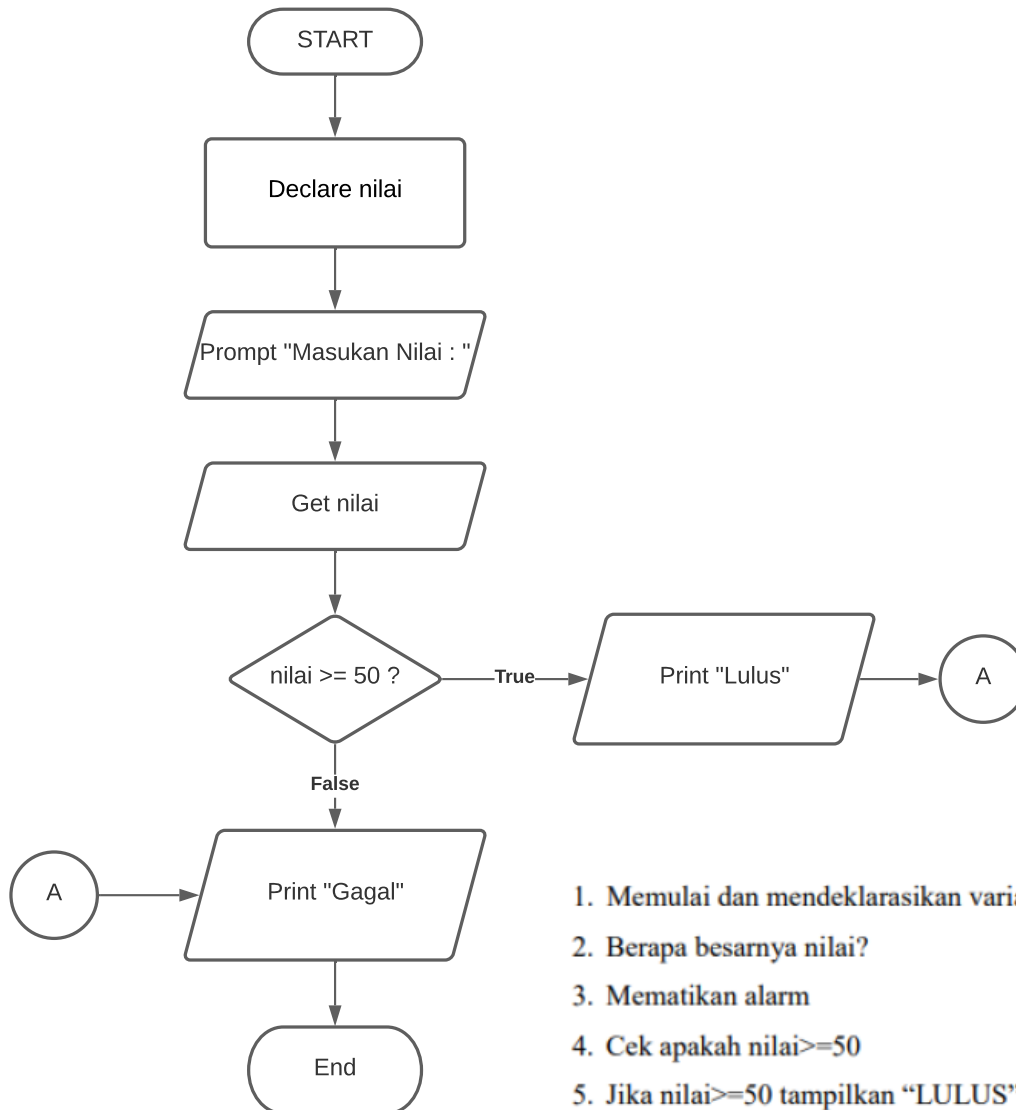
7. Sebutkan kelebihan dan kekurangan menggunakan modularisasi dalam pembuatan algoritma?

Solution:

Kelebihan: kodenya bisa dipakai kembali, mengurangi pengulangan penulisan, memudahkan dipahami, dan proses maintenance yang lebih efisien.

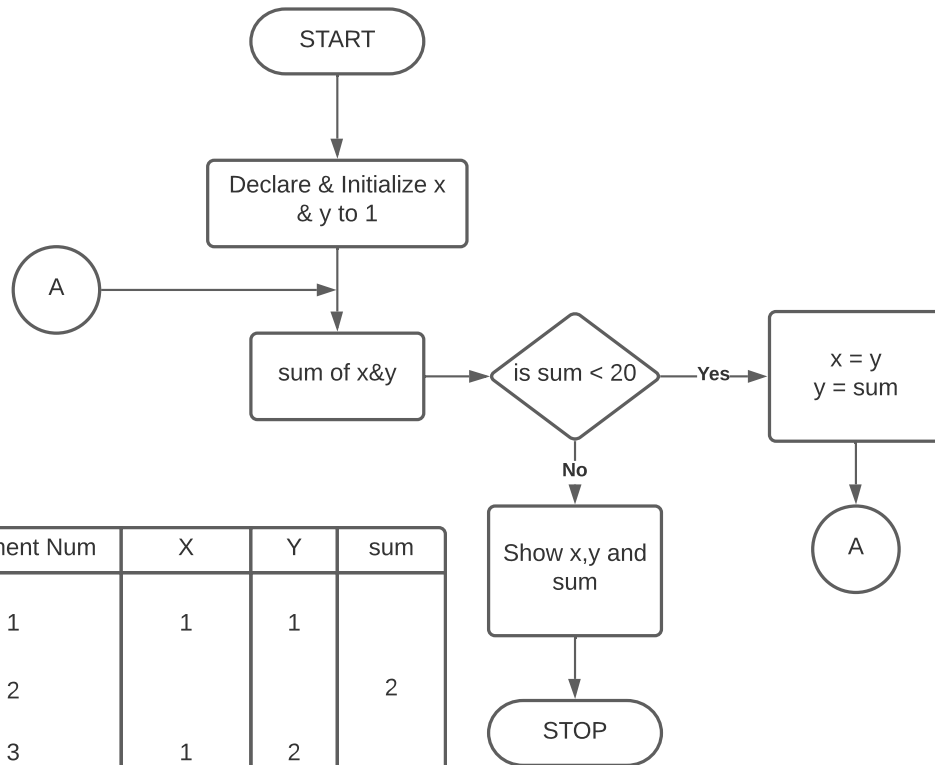
Kekurangan: jika dibagi terlalu banyak dan kecil akan kurang efisien

Lengkapi penulisan flowchart berikut ini



1. Memulai dan mendeklarasikan variabel nilai
2. Berapa besarnya nilai?
3. Mematikan alarm
4. Cek apakah nilai ≥ 50
5. Jika nilai ≥ 50 tampilkan "LULUS"
6. Jika nilai < 50 tampilkan "GAGAL"
7. Stop

Analisa flowchart dibawah ini dan selanjutnya lengkapi trace table!



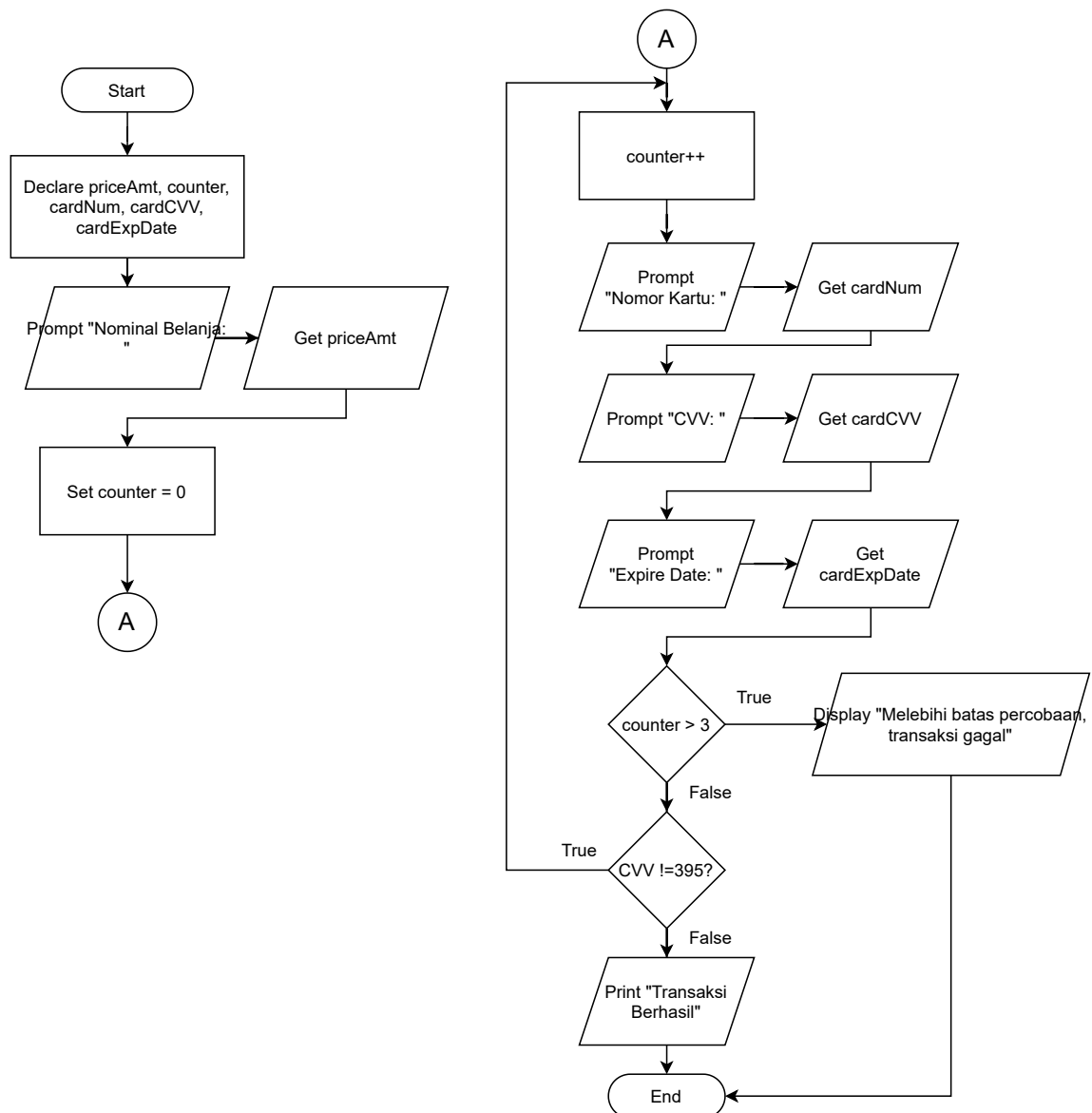
Statement Num	X	Y	sum
1	1	1	
2			2
3	1	2	
2			3
3	2	3	
2			5
3	3	5	
2			8
3	5	8	
2			13
3	8	13	
2			21
3	OUT	OUT	OUT

Flowchart dan Pseudocode

Problem 1

Defining Diagram

Input	Processing	Output
cardNum	Declare the card	priceAmt
cardCVV	Set counter the card	
cardExpDate	Insert card	



Pseudocode

- 1 Deklarasikan priceAmt, counter, cardNum, cardCVV, dan cardExpDate
- 2 Masukkan nominal belanja dan isi di variabel priceAmt
Lakukan:
 - 3 Tambahkan 1 kedalam counter
 - 4 Masukkan nomor kartu dan isi di variabel cardNum
 - 5 Masukkan CVV dan isi di variabel cardCVV
 - 6 Masukkan Expiration Date dan isi di variabel cardExpDate
 - 7 Jika counter > 3, maka:
 - Berikan error
 - Hentikan program
- 8 Print "Transaksi Berhasil!"

Desk Checking

Input	Dataset1	Dataset2
priceAmt	15000	15000
cardNum	860487957	860487957
cardCVV	395	364
cardExpDate	1025	1025

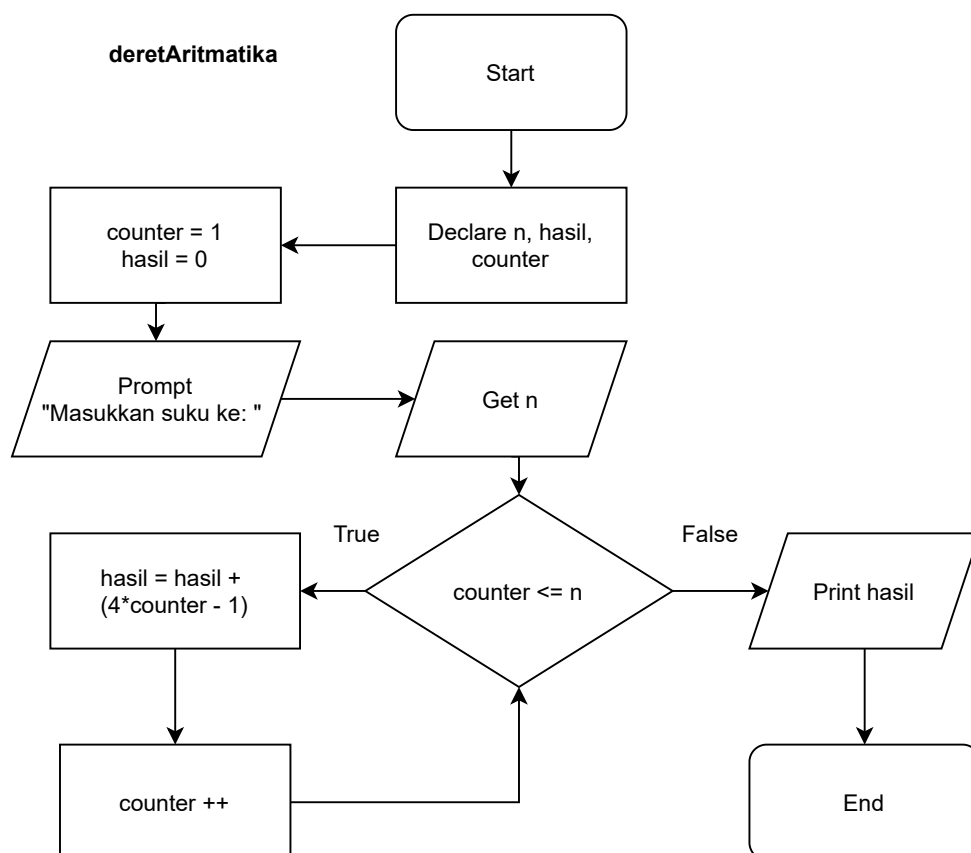
Expected Output	Dataset1	Dataset2
OUTPUT	Transaksi Berhasil!	Melebihi batas percobaan, transaksi gagal

Statement Num	priceAmt	cardNum	cardCVV	cardExpDate	counter	OUTPUT
First Pass						
1						
2	15000					
3					1	
4		860487957				
5			395			
6				1025		
7						
8						"Transaksi Berhasil!"
Second Pass						
1						
2	15000					
3					1	
4		860487957				
5			364			
6				1025		
7						
1						
2	15000					
3					2	
4		860487957				
5			364			
6				1025		
7						
1						
2	15000					
3					3	
4		860487957				
5			364			
6				1025		
7						
1						
2	15000					
3					4	
4		860487957				
5			364			
6				1025		
7						Melebihi batas percobaan, transaksi gagal

Problem 2

Defining Diagram

Input	Procces	Outuput
n	Get n Initialize counter & hasil Calculate hasil	hasil



Pseudocode

- 1 Deklarasikan variabel `n`, `counter` dan `hasil`
- 2 Isi variabel `hasil` dengan angka 0
 Isi variabel `counter` dengan angka 1
- 3 Masukkan nilai suku dan isi di variabel `n`
 Selama `counter` kurang dari atau sama dengan `n`:
 - 4 Isi variabel `hasil` dengan isi dari variabel `hasil`
 ditambah `hasil` dari rumus $(4 * \text{counter} - 1)$
 - 5 Tambahkan satu pada `counter`
- 6 Munculkan isi dari variabel `hasil`

Desk Checking

Input	Dataset1	Dataset2
n	3	5

Expected Output	Dataset1	Dataset2
hasil	21	55

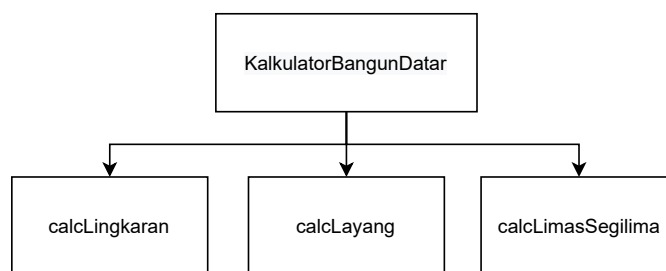
Statement Num	n	counter	hasil
First Pass			
1			
2		1	0
3	3		
4			3
5		2	
4			10
5		3	
4			21
5		4	
6			OUTPUT
Second Pass			
1			
2		1	0
3	3		
4			3
5		2	
4			10
5		3	
4			21
5		4	
4			36
5		5	
4			55
5		6	
6			OUTPUT

Problem 3

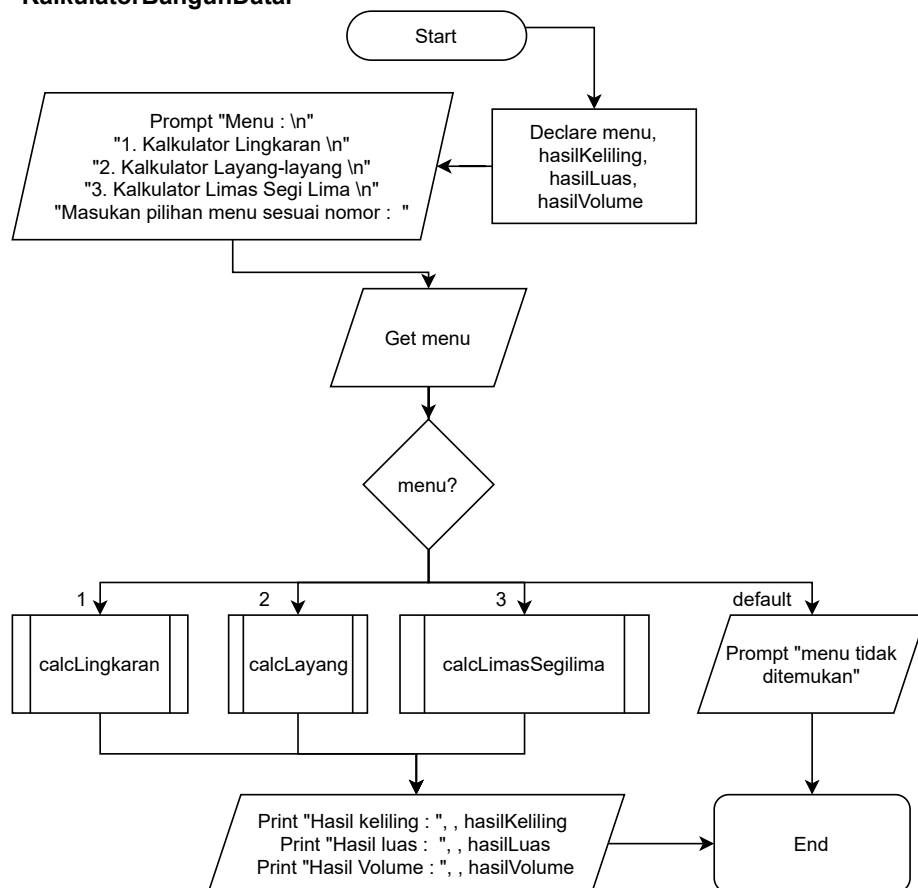
Defining Diagram

Input	Proses	Output
menu jari sisi1 sisi2 diagram1 diagram2 luasAlas tinggi	Get menu Get variable based on menu Calculate module Print result based on menu	hasilKeliling hasilLuas hasilVolume

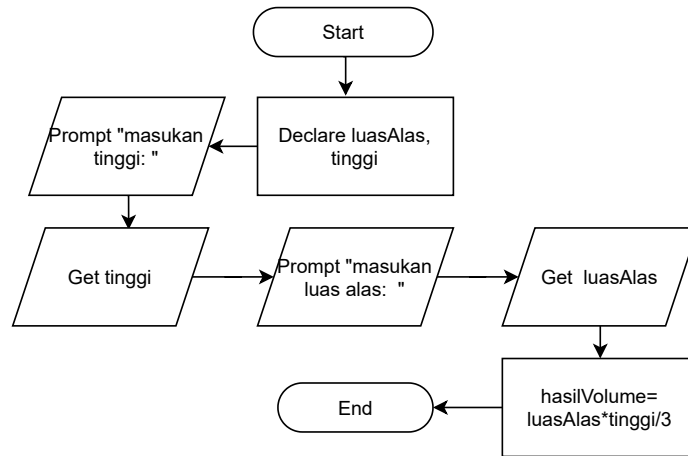
Hierarki Flowchart



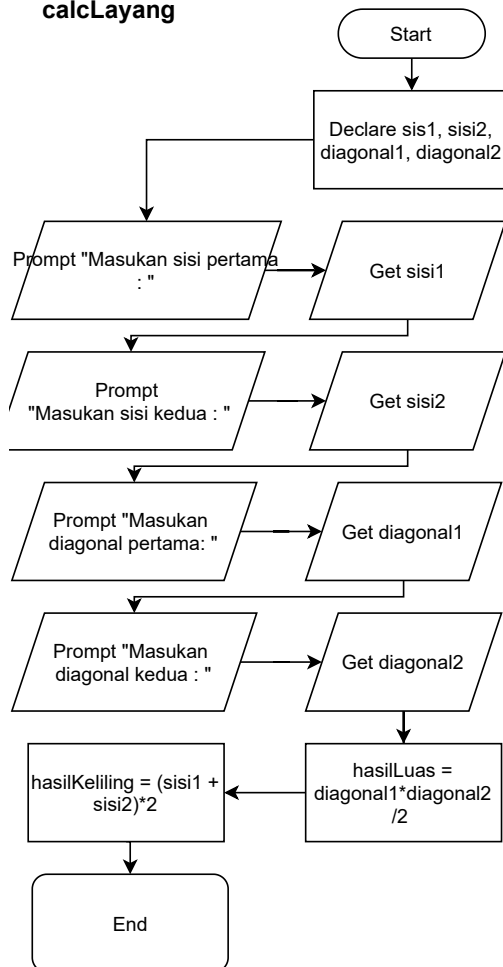
KalkulatorBangunDatar



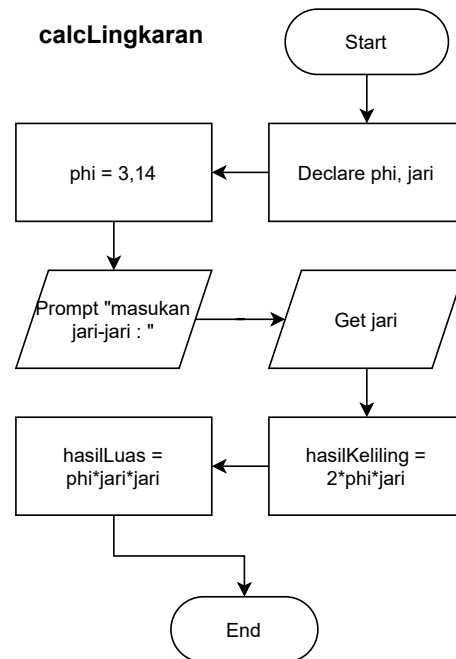
calcLimasSegilima



calcLayang



calcLingkaran



Pseudocode

kalkulatorBangunDatar

START

```
1  DECLARE menu, hasilKeliling, hasilLuas, hasilVolume
2  PRINT "1. Kalkulator Lingkaran"
   PRINT "2. Kalkulator Layang-layang"
   PRINT "3. Kalkulator Limas Segilima"
3  PROMPT "Pilih menu: "
   GET menu

4  CASE of menu
    1: calcLingkaran
    2: calcLayang
    3: calcLimasSegilima
    default: PRINT "Menu tidak ditemukan"
  ENDCASE

5  PRINT "Hasil Keliling: ", hasilKeliling
   PRINT "Hasil Luas: ", hasilLuas
   PRINT "Hasil Volume: ", hasilVolume
END
```

calcLingkaran

START

```
1  DECLARE phi, jari
2  phi = 3.14
3  PROMPT "Masukkan jari-jari: "
   GET jari

4  hasilKeliling = 2 * phi * jari
5  hasilLuas = phi * jari * jari
END
```

calcLayang

START

```
1 DECLARE sisi1 , sisi2 , diagon1, diagon2
2 PROMPT "Masukkan panjang sisi pertama: "
  GET sisi1
  PROMPT "Masukkan panjang sisi kedua: "
  GET sisi2
  PROMPT "Masukkan panjang diagonal pertama: "
  GET diagon1
  PROMPT "Masukkan panjang diagonal kedua: "
  GET diagon2

3 hasilKeliling = 2 * (sisi1 + sisi2)
4 hasilLuas = (diagon1 + diagon2) / 2
END
```

calcLimasSegilima

START

```
1 DECLARE luasAlas , tinggi
2 PROMPT "Masukkan luas alasnya: "
  GET luasAlas
  PROMPT "Masukkan tinggi: "
  GET tinggi

3 hasilVolume = (luasAlas * tinggi) / 3
END
```

Desk Checking

Input	Dataset1	Dataset2
menu	1	3
jari	10	
luasAlas		10
tinggi		3

Expected Output	Dataset1	Dataset2
hasilKeliling	62.8	
hasilLuas	314	
hasilVolume		10

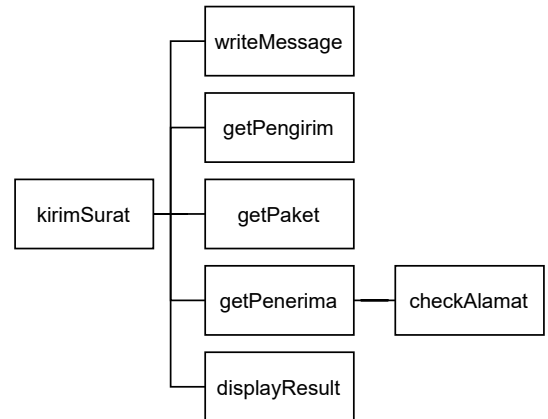
Statement	menu	phi	jari	luasAlas	tinggi	hasilKeliling	hasilLuas	hasilVolume
First Pass								
main()								
1								
2								
3	1							
4								
calcLingkaran()								
1.1								
1.2		3.14						
1.3			10					
1.4						62.8		
1.5							314	
main()								
5						OUTPUT	OUTPUT	OUTPUT
Second Pass								
1								
2								
3	3							
4								
calcLimasSegilima()								
1								
2				10	3			
3								10
main()								
5								OUTPUT

Problem 4

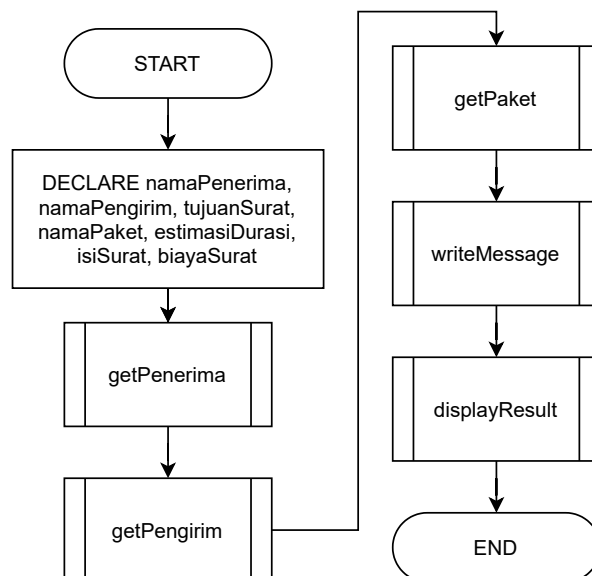
Defining Diagram

Input	Process	Output
namaPenerima	User menginput data penerima surat	namaPenerima
namaPengirim	User menginput data pengirim	namaPengirim
tujuanSurat	User menginput pilihan paket	tujuanSurat
pilihanPaket	User menginput pesan yang ingin dikirim	namaPaket
isiSurat	Program menampilkan hasilnya	estimasiDurasi
		biayaSurat
		isiSurat

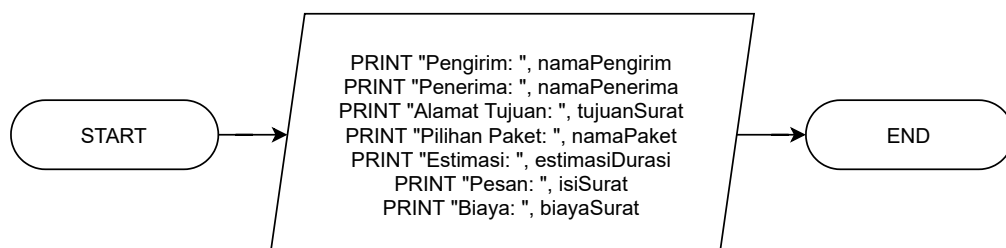
Hierarchy Table

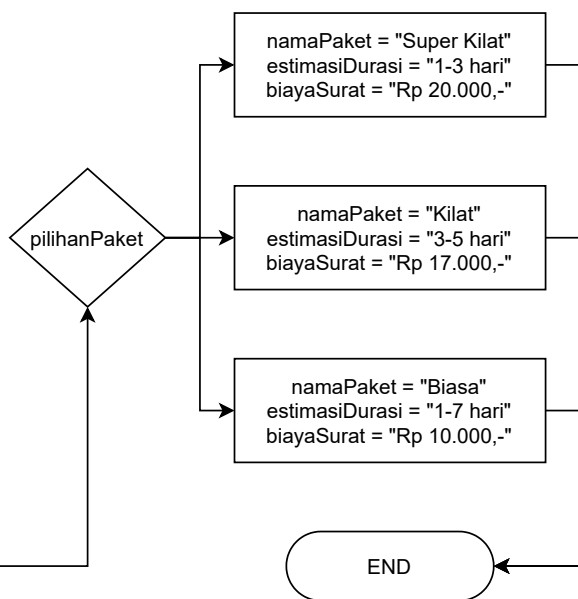
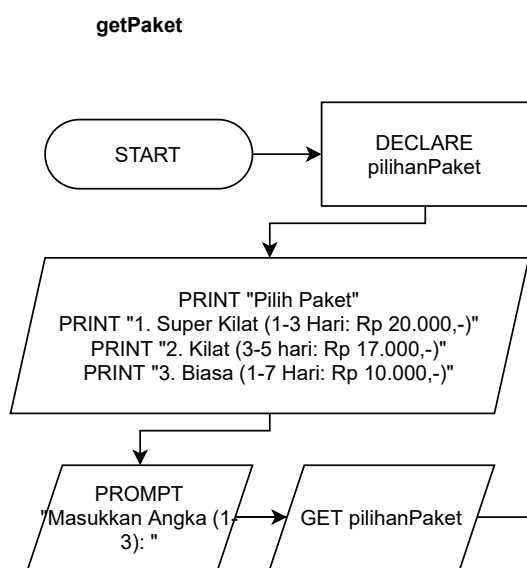
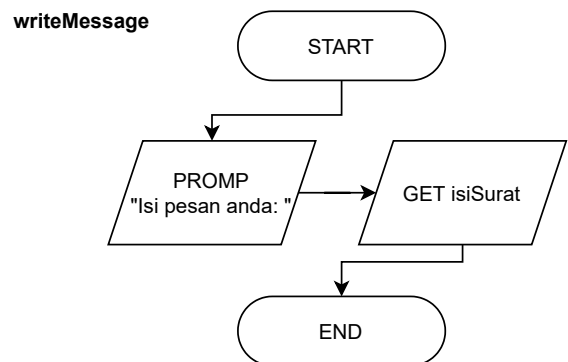
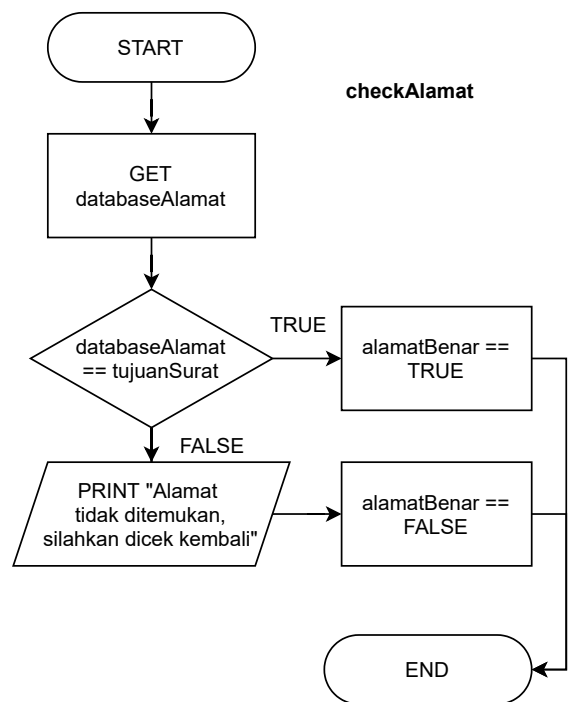
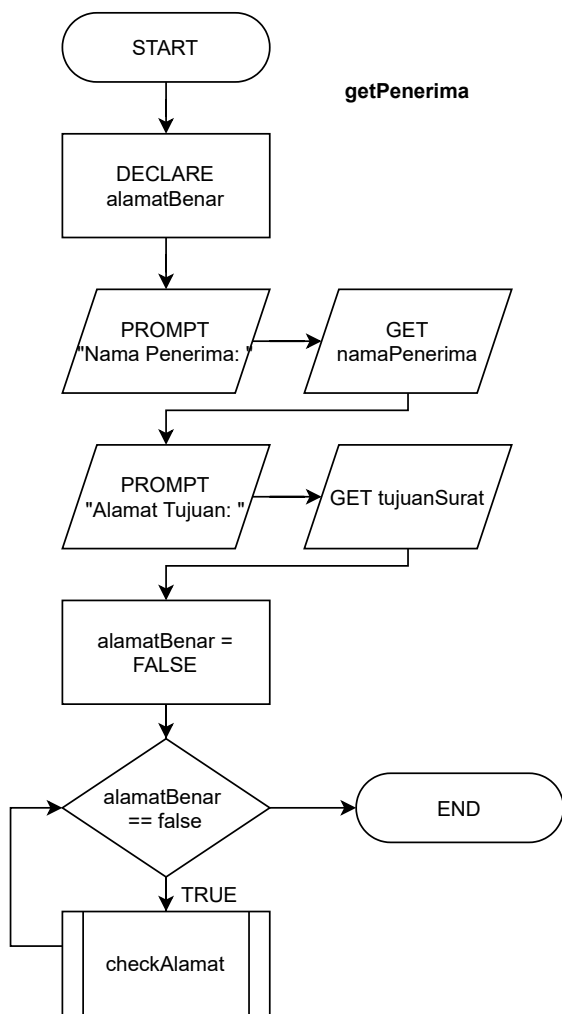


kirimSurat



displayResult





Pseudocode

kirimSurat

START

```
1  DECLARE namaPenerima, namaPengirim, tujuanSurat,
    namaPaket, estimasiDurasi, isiSurat, biayaSurat
2  getPenerima()
3  getPengirim()
4  getPaket()
5  writeMessage()
6  displayResult()
```

END

getPenerima

START

```
1  DECLARE alamatBenar
2  PROMPT "Nama Penerima: "
   GET namaPenerima
   PROMPT "Alamat Tujuan: "
   GET tujuanSurat
3  alamatBenar = FALSE
4  WHILE (alamatBenar == FALSE)
    checkAlamat()
   ENDWHILE
```

END

checkAlamat

START

```
1  GET databaseAlamat
2  IF (databaseAlamat == tujuanSurat)
    alamatBenar == TRUE
   ELSE
    alamatBenar == FALSE
```

```
        PRINT "Alamat tidak ditemukan, silahkan dicek kembali"
    ENDIF
END
```

getPengirim

START

```
1  PROMPT "Nama Pengirim: "
```

```
    GET namaPengirim
```

END

getPaket()

START

```
1  DECLARE pilihanPaket
```

```
2  PRINT "Pilih Paket"
```

```
    PRINT "1. Super Kilat (1-3 Hari: Rp 20.000,-)"
```

```
    PRINT "2. Kilat (3-5 hari: Rp 17.000,-)"
```

```
    PRINT "3. Biasa (1-7 Hari: Rp 10.000,-)"
```

```
    PROMPT "Masukkan angka (1-3: "
```

```
3  GET pilihanPaket
```

```
4  CASE OF pilihanPaket:
```

```
    1: namaPaket = "Super Kilat"
       estimasiDurasi = "1-3 hari"
       biayaSurat = "Rp 20.000,-"
```

```
    2: namaPaket = "Kilat"
       estimasiDurasi = "3-5 hari"
       biayaSurat = "Rp 17.000,-"
```

```
    3: namaPaket = "Biasa"
       estimasiDurasi = "1-7 hari"
       biayaSurat = "Rp 10.000,-"
```

END

writeMessage

START

1 PROMPT "Isi pesan anda: "

GET isiSurat

END

displayResult

START

1 PRINT "Pengirim: ", namaPengirim

PRINT "Penerima: ", namaPenerima

PRINT "Alamat Tujuan: ", tujuanSurat

PRINT "Pilihan Paket: ", namaPaket

PRINT "Estimasi: ", estimasiDurasi

PRINT "Pesan: ", isiSurat

PRINT "Biaya: ", biayaSurat

END

Desk Checking

Statement Number	namaPenerima	namaPengirim	tujuanSurat	alamatBena	pilihanPaket	isiSurat	namaPaket	estimasiDuras	biayaSurat
main()									
1									
getPenerima()									
1									
2	Agus		Jakarta						
3				FALSE					
checkAlamat()									
1									
2				TRUE					
getPengirim									
1		Rivo							
getPaket()									
1									
2									
3					2		Kilat	3-5 hari	Rp 17.000,-
4									
writeMessage()									
1						Lorem Ipsum			
displayResult()									
1	OUTPUT	OUTPUT	OUTPUT			OUTPUT	OUTPUT	OUTPUT	OUTPUT